Aalto University
School of Science
Master's Programme in Computer, Communication and Information Sciences

Inan Evin

Cine-AI

Emulating Director Styles in Game Cutscenes

Master's Thesis
Espoo, May 24, 2021

Supervisor:	Professor Perttu Hämäläinen, Aalto University
Advisor:	Dr. Christian Guckelsberger, Aalto University

Aalto University
School of Science
Master's Programme in Computer, Communication and Information Sciences

ABSTRACT OF MASTER'S THESIS

Author:	Inan Evin		
Title: Cine-AI Emulating Director Styles in Game Cutscenes			
Date:	May 24, 2021	**Pages:**	vi + 42
Major:	Game Design and Production	**Code:**	SCI3046
Supervisor:	Professor Perttu Hämäläinen		
Advisor:	Dr. Christian Guckelsberger		

We propose Cine-AI, a novel open-source toolset that allows mimicking famous movie directors' cinematography styles in in-game cutscenes. As the quality and production costs of such scenes increase, game companies and developers need new tools for automating the process. Such methods and tools have been proposed, but most of them lack empirical data of real directors, resulting in a lack of unique filmography characteristics. Cine-AI proposes a set of tools that use a director style description extracted from an annotated dataset of video clips by the director. This data is processed by Cine-AI, resulting in procedurally generated camera framing, movement and transitions. The final composition is presented to the users in design-time by the means of a storyboard, allowing full customization and fine tuning. We evaluate the proposed system through two experiments. First, we quantitatively confirm that Cine-AI can produce recognizable director styles, at least in the case of the two directors used in our study. Our participants were able to distinguish the generated Quentin Tarantino and Guy Ritchie styles with an accuracy of 79%. Second, we conducted a usability study with 12 game developers, resulting in a moderately high System Usability Scale score and qualitative comments for informing future work. For instance, the users appreciated Cine-AI's ease of use and suggested more documentation such as tooltips or interface wizards.

Keywords:	procedural, automated, camera control, procedural camera, emulating directors, director styles, procedural camera motion, cutscene, in-game cutscene, cinematics
Language:	English

Acknowledgements

I want to thank Professor Perttu Hämäläinen for his support and good guidance.

Espoo, May 24, 2021

Inan Evin

Abbreviations and Acronyms

GUI	Graphical User Interface
API	Application Programming Interface
ML	Machine Learning
SVM	Support Vector Machine
DCCL	Declarative Camera Control Language
JSON	JavaScript Object Notation, a file format
NPC	Non-player Character
PCA	Principal Component Analysis,a statistical analysis method
IRR	Inter-rater Reliability
ICC	Intraclass Correlation
SUS	System Usability Scale, a questionnaire
CTA	Concurrent Think-Aloud, an experiment protocol

Contents

Chapter 1

Introduction

In-game cinematic cutscenes are non-interactive sequences in a video game that break up and pause gameplay. Such scenes can be used to progress the story or to show conversations and events concerning the characters. They include character animations, scene composition and extensive cinematography, especially in high quality productions. During a cutscene, players are to watch the scene from a virtual camera's point of view. Game developers have the responsibility to handle the placement of these virtual cameras, along with their motion and behaviours in order to ensure to get the best shots possible depending on the cinematic feeling they aim to achieve. The process of governing the sequence's animations along with the camera can be classified as digital cinematography and directing [1]. That is why most game companies, especially those with a high budget, tend to hire directors and cinematographers along with movie productions teams in order to shoot and direct their cutscenes. Alternatively, it is also possible to use various techniques in attempt to automate the process of directing and virtual camera management. Acceleration of the said process for a game that includes considerable amount of cutscenes would reduce the development costs drastically. On the other hand, it might result in monotonous and repetitive scenes since an artificial intelligence would lack the input of a real director, which we can rephrase as the lack of a director's style.

In order to simulate the camera behaviour similar to a real director shooting the scene, there are numerous obstacles that a system needs to overcome. Problems such as camera placement, subject visibility, shot continuity and scene composition, along with creating a director feeling by imposing a unique shooting style have been partially addressed in a number of studies. For instance, Christie and Olivier [7] explain techniques and requirements for procedural camera control while taking cinematography techniques into account. The paradigm explained by the study can be used to create an autonomous system that is able to convey scene data with respect to cinematography rules. Another study relates to the problem of continuity and narrative construction in automated cinematography by presenting a discourse planning technique [20]. Meanwhile, some studies introduce broader systems that focus on creating full scene compositions, such as the model explained by He et al. [18] or the declarative camera language proposed by Christianson et al. [6]. These studies present solid techniques and methodologies to overcome some or most of the problems, meanwhile failing to address the lack of a director's style.

To solve the problem, we propose and evaluate Cine-AI, an automated camera generation toolset that is able to mimic a chosen director's cinematography style. We first analyze the movie clips of two directors in terms of different cinematography techniques. The analysis yields style description data that our system uses while procedurally generating camera framing, movement and transitions. Cine-AI provides a storyboard tool to view the results of the generated scene, completely in design-time. The camera placement, transitions as well as the cinematography techniques used in the generated result are aimed to be as close as to the chosen director's style, providing uniformity and the characteristics most systems lack of. The user controls Cine-AI by indicating significant content on an animation timeline, e.g., by marking a point in time where specific objects need to be simultaneously visible. The user can also freely adjust the generated results using the storyboard view.

This paper makes the following contributions:

- A novel interactive system and toolset that solves the problem of mimicking a target director's shooting style, without using expensive machine learning computations or high-cost runtime processing.

- An open-source implementation of the toolset in the popular Unity 3D game engine, this enables both wide practical applicability and future research. The code is available at `https://github.com/inanevin/Cine-AI`.

- A user study (N=18, within-subjects) that measure similarity of our system's output with the sample directors' clips. Our participants were able to distinguish between Quentin Tarantino and Guy Ritchie given outputs generated by our toolset with 79% accuracy, confirming that Cine-AI can produce recognizable director styles.

- A user study (N=12) aimed at measuring the usability of each tool contained in our system using a standardized usability questionnaire.

Cine-AI scored B in a plus-minus letter grading system, indicating it can be used in production without any major usability flaws.

Chapter 2

Background And Related Work

The amount of material in the field of simulating a director's style in the context of games is scarce. However, we can find multiple studies that focus on a broader aspect of procedural camera generation. These studies revolve around the ideas of creating systems that are able to interpret virtual scene data and generate a meaningful output that can be used to create scene compositions, including camera placements and cinematography techniques. They mostly focus on overcoming individual problems such as the automation of camera placement, subject visibility, scene continuity and narrative progression. There is also some, albeit much more limited, research on creating storyboards in order to utilize these notions in a design-time based manner. Below, we review some of the previous studies briefly in the context of problems mentioned above and discuss their key points.

2.1 Camera Placement

There are numerous techniques studied for placing a virtual camera within a 3D scene. Arijon [2] describes number of idioms for cinematography. For example, in dialogue sequences, the camera should be placed within the field-of-view cone of the subject in a single character environment. However for multiple characters, it is necessary to create a line of action using the middle points of the characters' positions to invoke the triangle principle. Christie and Olivier [7] discuss different models of digital cinematography that amplifies the importance of such idioms. He et al. [18] describe a set of heuristics and constraints used in their Virtual Cinematographer that tend to submit to broader approaches and idioms defined by film makers throughout the years. In this thesis, Arijon [2]'s idioms, rules of thumb such as culling heists mentioned by He et al. [18] for camera placement, along with most common filmography approaches to create camera angles are used when initially determining a specific position for a particular shot.

2.2 Subject Visibility

A significant concept related to camera placement is subject visibility. In a 3D cutscene, it is common to have many virtual entities, characters, objects and even particle effects dynamically being simulated around the environment. An automated system responsible for camera management needs to ensure a clear shot of the target subject. Visibility volumes proposed by Lino et al. [26] address the issue by generating a 2D *cell-and-portal* representation of the 3D environment. Within this abstraction, the scene is divided into cells and portals in order to determine key areas for subject visibility. Meanwhile Oskam et al. [30] propose an algorithm for visibility-aware path-planning in a virtual environment. Pair-wise visibility data for various parts of the scene along with a pre-computed representation of collision-free paths are used in order to execute a camera transition with clear subject focus during runtime.

In Cine-AI, a similar approach to Lino et al. [26] is used but completely within 3D context. We analyse the current scene from the camera's point of view and perform world-based line tracing to calculate visibility volumes. Similarly to Oskam et al. [30], we use design-time calculated *Scene Proxies* to ensure a collision-free camera motion in runtime.

2.3 Continuity and Cinematography

A multitude of works address the issues of creating meaningful transitions and achieving shot continuity in a virtual environment. One method to achieve this is to abstract the whole animation timeline with various states, each state having a particular precondition or a goal to achieve [20]. These can be anything in the context of the application, such as looking at a particular subject or precondition of some game event being triggered. Then the whole scene is treated as a one big state machine, generating shot sequences and motion plans to choose the best result according to the current state. Jhala and Young [20] use this technique along with a number of parameters including shot significance to rank the generated sequences. These sequences are then sorted with a best-first approach in order to select the best result. In such systems, link planning systems like Longbow by Young and Moore [39] can be used extensively to generate these sequences. Similar to Jhala and Young [20], Cine-AI uses various user-given parameters that determine a shot's importance, pace and action value for ranking. However in contrast to most other techniques, the best possible shot is not auto-selected by the system in runtime, rather it is offered to the developer in design-time in the means of storyboards. Thus, we avoid the computation of discourse planning links during the game while also providing a complete flexibility to the designer of the scene. Additionally, Cine-AI relies on the usage of cinematography techniques to ensure scene and shot connectivity, by complying with the rules derived from the best filmography practices. Thus, each shot generated by Cine-AI undergoes a series of checks and rules including comparisons with the previous shots to be considered as eligible.

Christianson et al. [6] propose a declarative camera control language (DCCL) in order to ratify some of the cinematography idioms into a more formal context. By formalizing the idioms, it becomes possible to create automated scenes that comply with particular cinematography techniques and rules. Additionally, it becomes easier to mimic a director's input as the formalization relies on the idioms most directors have been using in their works. A hierarchical representation of film idioms enables achieving the said formalization meanwhile making it easier to categorize rules of a scene along with the goals of particular shots. A similar representation is used by Jhala and Young [20] in order to present storytelling plans in a declarative manner. Therefore, along with selecting the best shot sequences, it becomes possible to create uniformity throughout meaningful transitions that convey the narrative elements properly. Concerned with idioms and rules, Karp and Feiner [21] and similarly Drucker and Zeltzer [10] suggest creating film grammars, coded with cinematography idioms, to generate a set of shot sequences using a top-down analysis. Generating such shots allows the calculation of motion planning approaches that can achieve certain visual goals. However, such systems are most suitable for non-dynamic virtual scenes as they depend on the timing information of the animations and would not react to dynamic changes in games.

Instead of solely focusing on film idioms, Cine-AI uses real director data, derived from hundreds of movie clips of a chosen director, in order to create meaningful scene composition. We use the data extracted from movie clips in an hierarchical manner, which then help our algorithm to determine the best cinematography technique to use depending on the user's choice of director and scene parameters. Since the chosen techniques have already been derived from a meaningful context, the directors' data, the generated shot sequences, along with the shot transitions make sense within the composition.

In order to select the best possible cinematography techniques and shot sequences for a particular time during an in-game cinematic cutscene, machine learning (ML) methods are also used in some studies. Soares de Lima et al. [35] propose a system that uses support vector machines (SVM) in order to figure out best possible shot selection with respect to cinematography techniques. A prediction process that takes the scene type, actor features and an SVM database into account is used to decide best possible outcome of a shot given a scene with known environment settings and number of actors. Although machine learning tools can also be used to produce the final result, this would not allow fine-grained control over the generation by the user. In contrast, Cine-AI only uses machine learning in the statistical analysis of director data, but the procedural generation rules and parameters can be adjusted by the user.

2.4 Storyboards

Although there exists previous work on storyboard generation, or even the extension of storyboards in live-action movie production context [17], studies on implementing storyboards for procedural 3D cinematography purposes are lacking. Ronfard et al. [31] propose a storyboard language that describes each shot in terms of a sentence, which can be used to build software systems that convert formal shot descriptions into visual storyboard panels. This provides a way to automate the storyboarding process as well as virtual directing. In contrast, Cine-AI uses storyboards only as a means of visualisation and editing tool. In Cine-AI, the produced output scene is represented in a storyboard interface. This interface, acting as a summery of the user's scene, provides functionality to regenerate one or multiple shots, along with the options to adjust the generation parameters. To the best of our knowledge, no such interface has been developed previously.

Chapter 3

System

The interaction and processing steps of Cine-AI can be summarized as follows, with more implementation details and design rationale provided in the sections below.

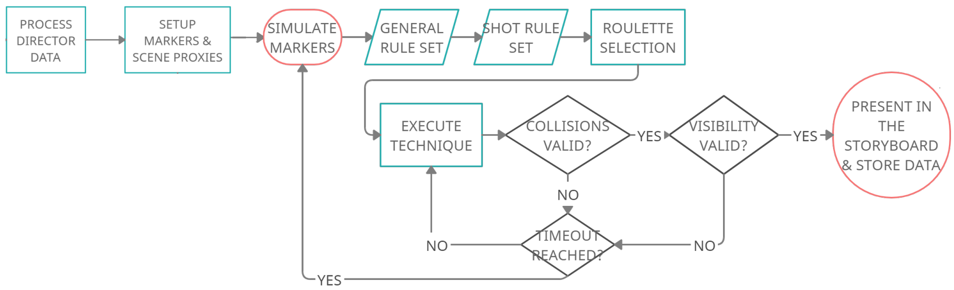

Figure 3.1: UML diagram demonstrating the simulation process.

- Users are prompted to input director style data that Cine-AI will try to imitate and help define the cinematography rules to abide by while calculating camera behaviour. To generate such data, we developed a video annotation and feature extraction approach that we tested using a total of 160 clips from Quentin Tarantino and Guy Ritchie.

- *Shot markers* are used to define cuts and transitions. The user places the markers at desired position on a timeline view.

- Users are also given a set of parameters to tweak the behaviour of the toolset depending on their project needs. These parameters mostly define 3D scene specific settings like camera distances, allowed field-of-view ranges and geometry analysis.

- Cine-AI collects static collision data from the cutscene by calculating scene proxies, 3D volumes that are used to define the boundaries of the cutscene.
- Users can simulate all or specific shot markers over the cutscene. Simulation of a marker triggers the system to choose a specific cinematography techniques for the marker's point in the timeline. A new camera behaviour, including the camera placement, angle, path, alignment and lens parameters is calculated based on the chosen techniques.
- After all shot markers are simulated, the calculated data is presented to the user in the form of a storyboard, where each marker will correspond to a single node, along with a preview of the calculated shot. Users can individually tweak the calculated data or re-simulate completely if they desire.
- After a final scene composition is achieved, users simply trigger their cutscenes during the game and notify Cine-AI's event system, which controls all camera behaviour in runtime based on the data that was calculated during design-time.

In order to allow easy utilization of Cine-AI in real-life game production, we have implemented it in the Unity 3D game engine that provides extensive set of tools for cutscene production, such as sequence and timeline editors, as well as immediate GUI libraries.

3.1 Director Data

In order to simulate director styles, our toolset needs cinematography data in an interpretable format. This section describes the analysis and extraction of the said data.

3.1.1 Choosing Directors

For the purpose of this research, we have chosen Quentin Tarantino and Guy Ritchie as target directors. There are two main reasons behind this choice.

- Both directors have recognizable and unique cinematography styles. This made it reasonable to hypothesize that emulating the styles would be possible in this proof-of-concept study.

- Both directors frequently employ cinematography techniques that are implementable and distinguishable on their own in the means of camera management. Many other techniques are dependent on post processing, audio and visual effects, which are beyond the scope of this research.

3.1.2 Coding the Data

We have selected 15 cinematography techniques to implement in Cine-AI. These techniques are as follows:

- **Cut**
- **Shots**; Extreme, Long, God's Eye, Medium, Master, Close-up, Free and Pan shot
- **Zooms**; Close-up, Quick and Dolly zoom
- **Tracking**; Handheld and Steadycam tracking
- **Slow-motion**

As indicated by the list above, we have implemented various types of shot, zoom and tracking techniques. On the other hand, we have only used a single type of cut technique for all transitions since most other techniques such as L Cut, Jump Cut or Cross Cut have more to do with post-process editing than the camera management.

80 different short clips for each director were analyzed in regards to the selected cinematography techniques. For the 160 clips in total, we coded the frequency of occurrence of all techniques. Moreover, two additional variables were coded into a numeric range between 0 and 1 in order to assess the dramatization and the pace of the analyzed scene. Dramatization refers to the emotional intensity of the scene in regards to character reaction, meanwhile pace represents how fast the scene's narrative unfolds. These values allow Cine-AI to better characterize a director's cinematography style by coding more information about the relation between the usage of a cinematography techniques and the type of the scene. Thus, Cine-AI can use the target director's dramatization and pace thresholds to sort the cinematography techniques based on the dramatization and pace values provided by users for their own cutscenes.

3.1.3 Analyzing Feature Importance

It is important to make sure the cinematography techniques we selected were contributing features when distinguishing between example directors. Given so, Cine-AI would be able to generate recognizable output when a different director is given as input. To ascertain the discriminative power of the cinematography techniques, we trained a logistic regression model to predict the director of each annotated video clip, using the annotated cinematography technique frequencies as the regression features. Logistic regression models the probability of a director as proportional to $\sigma(wTx+b)$, where σ is the logistic sigmoid function, x is a vector of features, w is a vector of feature weights, and b is an optional scalar parameter. The absolute value of a feature weight can be interpreted as the feature's estimated importance in correctly predicting the director. If a feature is not at all predictive of a director, logistic regression will assign it a zero weight.

Technique	**Weight**	**Technique**	**Weight**	**Technique**	**Weight**
God's eye	0.8398	Dolly zoom	0.4125	Free shot	0.1480
Steadycam tracking	0.6114	Close-up zoom	0.3903	Pan shot	0.1341
Handheld tracking	0.5922	Slow-motion	0.3355	Master shot	0.0671
Close-up shot	0.5315	Medium shot	0.3098	Stationary	0.0458
Quick zoom	0.4837	Long shot	0.2773	Cut	0.0151

Table 3.1.3: Logistic regression weights for each feature (technique) representing their contribution in distinguishing between Tarantino and Ritchie.

The logistic regression model is able to predict the correct director with an accuracy of 83.75%, providing evidence that the selected features are indeed able to characterize the director styles, at least for our selected directors. The feature weights are shown in Table 3.1.3, each indicating how distinguishable the techniques are. For instance, a god's eye view is commonly used by Quentin Tarantino but less frequently by Guy Ritchie. Ritchie also uses steadycam tracking more often (24.6% of our clips) than Tarantino (12.4% of our clips). On the other hand, overall cut/transition frequencies do not differ significantly between the directors (786 occurrences for Ritchie, 895 for Tarantino). Despite some regression weights being closer to zero, all the techniques were fairly effortless to implement. Moreover, even though they do not distinguish between our example directors, it's possible that they might be relevant for others. Thus, we implemented all the techniques in Cine-AI.

3.1.4 Data Classification

Not all the cinematography techniques used in the statistical analysis define the same type of camera behaviour. For instance, a close-up shot requires placing the camera closer to the subject's face [28], meanwhile the quick zoom technique focuses on re-adjusting the camera's lens field of view towards a target [38]. Due to the differences of execution between the techniques, it makes sense to collect them into similar categories in order to create a meaningful implementation. Thus, they were placed into 4 distinct categories, given in Table 3.1.4. Each category contains a default technique to fall back if our system can not determine a specific technique to use later on in the simulation.

Category	**Techniques**	**Default**
Positioning	Close-up, god's eye, master shot, pan shot and free shot	Free shot
Look	Quick zoom, dolly zoom, close zoom and stationary shot	Stationary shot
Track	NoTrack, steadycam variations, handheld variations	NoTrack
FX	NoFX, slow-motion	NoFX

Table 3.1.4: Cinematography categories and techniques within.

The *positioning* category defines where to place the camera within the 3D geometry. After the camera is placed in the scene, it always orients towards the target subject defined by the user. The camera's follow up behaviour, whether there will be changes in the lens parameters or not after the orientation is denoted by the *look* category. Next, *track* category defines the camera movement such as tracking a subject or imitating a walk-bob during the time between two shot markers. Finally, *FX* category is used to include any other cinematography techniques that does not fit into the above categories. Since our study does not focus on post effect production, only one technique is implemented for this category; slow-motion.

3.1.5 Processing The Data

In order to choose a cinematography technique for a shot marker in the timeline, Cine-AI relies on the probabilities of occurrence of each technique, coded during the director analysis stage. When a director's data is imported into Cine-AI, the probability of each cinematography technique with respect to their categories is calculated as:

$$P_x = f_x / \sum_{x}^{n} f_x$$

where f_x is the total frequency of technique x over all observed clips and n is the total number of techniques in the category.

Figure 3.2: Categorized director data showing default (P_x) for the cinematography techniques.

3.2 Shot Markers

In order for Cine-AI to simulate a scene composition, users are requested to define shot markers. By placing shot markers throughout their animation timeline, users can define which object(s) inside the game world they want the camera to focus on, along with the dramatization and pace values of the marker. By evaluating all markers, Cine-AI will decide on a new cinematography technique per marker, thus a new camera framing, movement and transition, resulting in a cut. Each marker will correspond to a node in the generated storyboard, which will be further explained later.

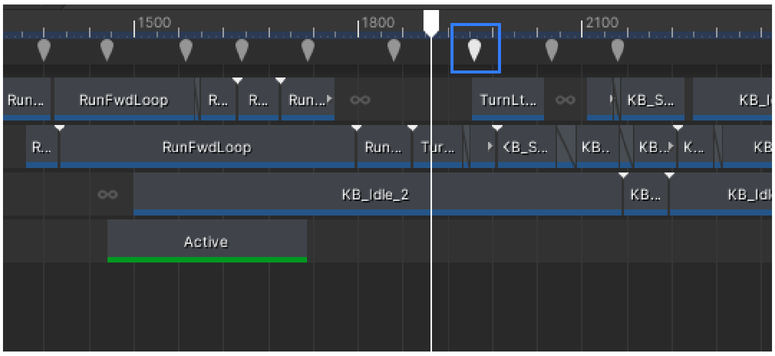

Figure 3.3: Marker implementation in Unity 3D game engine's animation timeline. Each marker (shown by the blue rectangle) marks a specific timestamp during the sequence and has options to define the target object at that timestamp.

3.3 Selecting Cinematography Techniques

During the simulation, Cine-AI selects multiple cinematography techniques to use at the timestamp of the marker. We use rejection sampling with technique sampling probabilities derived from the director style data and unsuitable techniques pruned based on hand-crafted rejection heuristics. This procedure is performed consecutively for all categories, with the order as follows: positioning, look, tracking and fx. Following a specified order allows Cine-AI to consider the decisions given in the previous categories while choosing a cinematography technique for each category. This section explains the decision process and the reasoning behind it.

3.3.1 General Rule Set

Cine-AI tries to abide by a set of general rules about cinematography in order to realistically compose scenes and process shot sequences. The said rules are considered as the ABCs of film-making. Some examples are:

Triangle Configuration: Arijon [2] suggests that in the case of multiple subjects within view, the camera should be focusing in the middle of an triangle created within the subjects.

Rule of Thirds: Rule of thirds is the process of creating natural balance in the shot composition by placing the subject on top of the cross-over points between imaginary lines [19]. Cine-AI tries to enforce the rule of thirds on target subjects, whilst allowing the users to control this process with exposed parameters. However, the visibility of the subject is always prioritized over the rule of thirds.

Leading Subjects: Any tracking camera should come to rest before the target subject stops during a continuously moving sequence.

Line of Action: An imaginary line of action connecting the major subjects in the scene shall be used during the calculation of camera positions.

Each cinematography technique within a category is processed through the general rule set to see if any technique should be deemed ineligible. For instance, if the 3D geometry composition at a particular time, defined by the marker, disallows a long distance shot, then techniques like long shot or master shot are eliminated. Likewise, if it's not possible to focus on multiple targets defined by a single marker at a time, all targets except the first one defined by the user are eliminated from the target list.

Abiding by all the rules hundred percent would not be feasible due to the interactive nature of the games and variable complexity of 3D scenes. Thus, Cine-AI provides controls for the users to bend the rules and let the system be as flexible as possible in their own projects. Exposed parameter sets for each technique, such as minimum and maximum shot distances, obedience thresholds to rule of thirds and similar idioms as well as the visibility checking options can be used to define how strict Cine-AI should behave during the decision process.

3.3.2 Shot-based Rule Set

Similar to the general rule set, a secondary rule set defining rules about shot compositions is used to further eliminate infeasible cinematography techniques. After a marker is processed with respect to the general rule set, Cine-AI looks at the techniques decided for previous categories, as well as the techniques decided by the previous marker. This information is incorporated in the decision process of the currently evaluated marker. The shot-based rule set includes the following key restrictions:

- It is not possible to use consecutive fast zoom techniques (quick zoom, dolly zoom).
- It is not possible to transit a master shot into a close-up shot. The camera should not be covering distances larger than a user-defined threshold at a single transition.
- It is not possible to use consecutive slow motion effects.

- It is not possible to apply any tracking technique that affects the camera position if dolly zoom is to be used in the Look category of the current marker.

These rules are mostly derived from well-accepted rules of cinematography. Additionally, they allow Cine-AI to avoid unwanted repetitions. In order to implement both general and shot-based rule sets, we used methods inspired by the Shotmaker of Kennedy and Mercer [22], where set of rules direct the possible outcome cases, further eliminating the available cinematography techniques until only a handful is left.

3.3.3 Using Director Data

After the rule-based analysis of particular marker is done, Cine-AI is left with possible cinematography techniques to choose from. This is where the director data comes into play. A simple roulette selection function using the calculated probability of the techniques is performed in order to choose an available technique in the particular category.

Once a technique is selected, the dramatization and pace values for the selected technique in the director's data is compared to the values designated in the marker. The comparison procedure determines whether the current marker is suitable to use the chosen technique. Users are also given threshold parameters that define how strongly the marker's dramatization and pace values should influence the calculation. Thus, users can technically skip the whole dramatization and pace calculation step and let the system choose the technique right away out of the roulette function.

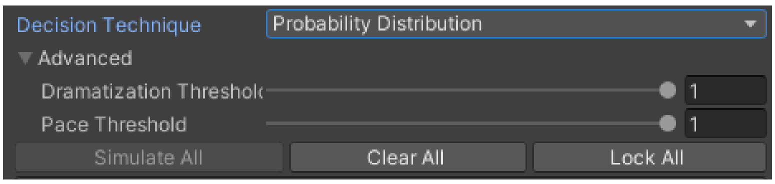

Figure 3.4: User interface showing user choices for decision function to use as well as dramatization and pace thresholds.

If a particular technique is chosen according to distributions but does not fit the dramatization and pace requirements, the process to choose a technique is repeated until a suitable technique is chosen or a timeout is reached. When the timeout occurs, Cine-AI selects the default technique for the category.

3.4 Scene Proxies

To prevent camera clipping and achieve collision-free camera paths, it is necessary to obtain 3D collision information from the geometry. In this section, we introduce the concept of scene proxies, which are 3D volumes that the users can use to define the area that the cutscene is taking place in. Cine-AI's interface provides parameters to adjust proxy settings per cutscene.

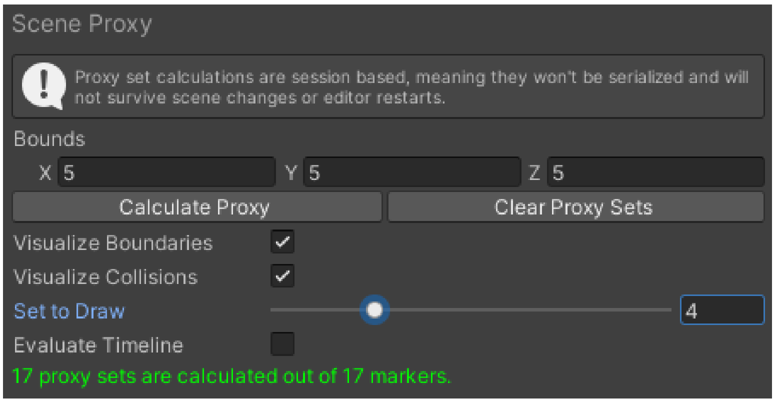

Figure 3.5: Exposed proxy parameters allow the users to compute proxy cells with an accuracy of their choice.

These settings define the boundaries of the proxies and their collision accuracy, and can be manipulated any time by the user.

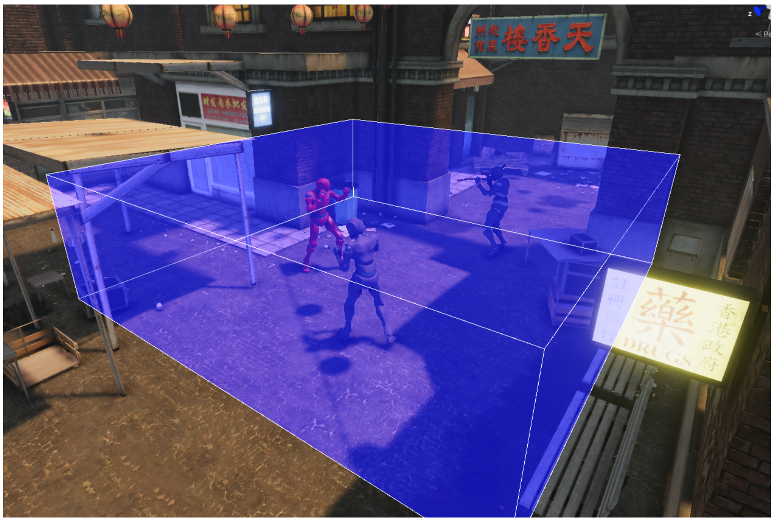

Figure 3.6: An example proxy defined by the user. The proxy is set to calculate all collision within the area.

Once the user set up scene proxies, the collision data can be calculated in design-time. Cine-AI goes through each shot marker, evaluating the cinematic sequence at the exact time of the marker so that the objects being animated will get in their respective poses for that time. Evaluation is followed by calculating the collision data for each object within the proxy, which is repeated for each marker in the timeline.

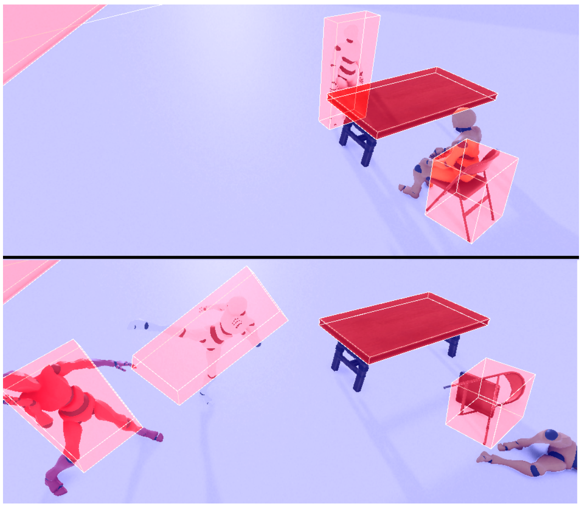

Figure 3.7: Example proxy calculation for 2 different markers at different times over the same cutscene. Each object containing a collision shape is marked with red.

Cine-AI serializes the calculated collision data which can later be used while selecting camera positions as well as during the game while performing runtime collision avoidance.

3.4.1 Simulation Process

So far, Cine-AI would have selected a cinematography technique from each category and obtained the collision data from the scene geometry. Next step for Cine-AI to perform is to find a suitable camera position based on the technique selected from *positioning* category, and stamp any *look*, *tracking* as well as *fx* behavior to be simulated in runtime. For each shot marker designated by the user, firstly the selected *positioning* technique is executed. The camera is placed according to the selected technique's implementation. Such implementations might dictate the camera shall be within a meter from the target object (close-up) or the camera shall be in such a distance it will have an overview of the whole scene and the target (master shot). After the camera is placed on a particular point in the geometry, the collision data from the scene proxies are used to determine whether the placement is valid or not. If deemed invalid, Cine-AI tries to find a new point based on the randomization properties of the executed technique until one is found or a timeout is reached. The camera will be oriented towards the subject point and visibility checks will start upon finding an eligible camera position. Cine-AI will perform an algorithm similar to line-casting, where a virtual capsule of adjustable radius will travel from the camera position towards the target position, checking if there are any objects blocking the path in between. If an occluder is detected, Cine-AI falls back to finding a new camera position and restarts the cycle. If the timeout is reached and no suitable position for the technique is found, Cine-AI will fall back to the technique selection process, selecting a new technique for the current category and restarting the simulation steps.

As mentioned previously, the execution cycle is only performed for the *positioning* category. Due to the dynamic nature of the games it is possible to have a multitude of animation and sequence possibilities during a cut-scene. Thus, continuous camera behaviors such as a quick zoom or handheld tracking techniques are executed in run-time.

3.4.2 Adjustable Parameters

All our implementations for cinematography techniques abide by the most common filmography rules [2]. For instance, Mascelli [28] explains that a close-up shot depicts the subject from chest to above the head, which our system tries to achieve for all close-up shots. However, not all the users are expected to have the same project needs. For instance, a user might have non-humanoid characters or a 3D environment based on abstract designs. Instead of imposing restrictions derived by obeying the cinematography rules, Cine-AI offers options for each cinematography technique to ensure maximum flexibility per project.

Figure 3.8: Example of per technique parameters exposed for users.

3.5 Storyboards

Cine-AI provides the users with an interface in order to edit and tweak the final scene composition. This section introduces the concept of storyboards, a flexible interface used for controlling the simulation process of procedural directing, motivated by the prevalence of storyboarding of movie pre-production.

A storyboard window is the central user control for creating scene compositions in Cine-AI. It contains controls for engine specific data and is responsible for displaying information such as imported director data, scene proxy settings and simulation parameters. Most importantly, it visualizes the final composition of the scene by displaying each timeline marker as an individual node, with previews of the camera angles selected.

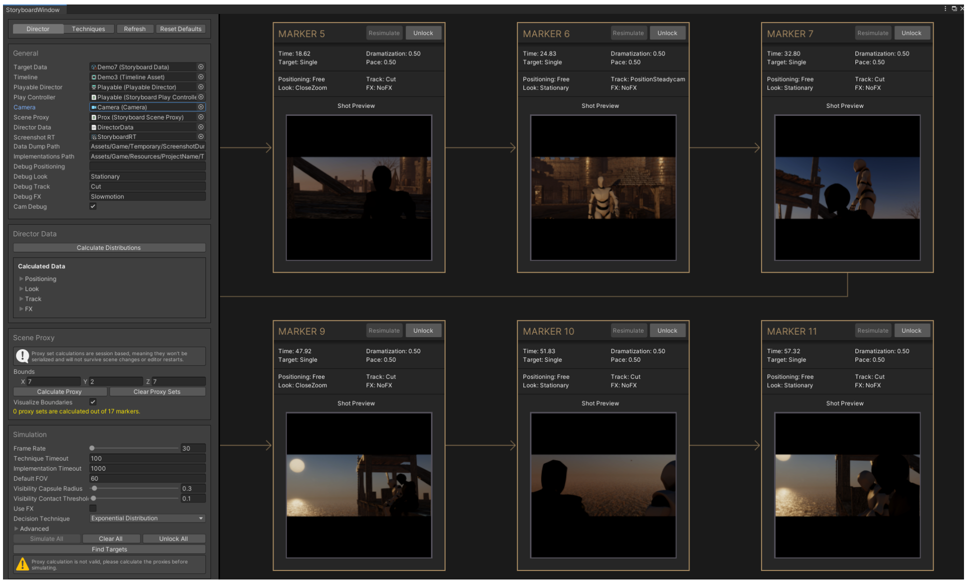

Figure 3.9: Storyboard window showing each simulated shot marker's data and previews.

Using the storyboard window, users are able to customize parameters related to the scene proxies, simulation and technique implementations. A preview of the results is shown for each marker and users can regenerate the results for one or more markers if not satisfied. Each marker has a lock/unlock option that determines whether it's protected from the changes while regenerating.

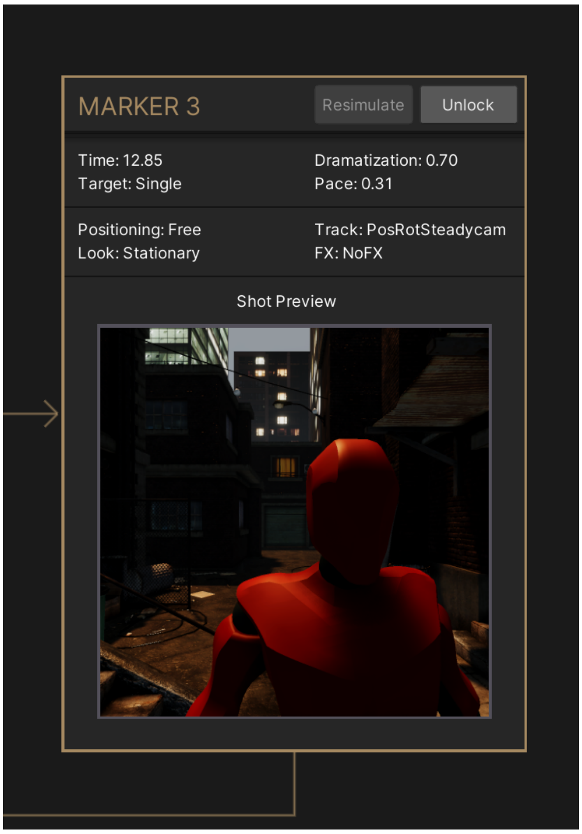

Figure 3.10: A storyboard node displaying a shot marker's parameters, selected cinematography techniques as well as a preview of the marker.

By utilizing the full power of the storyboard window, users have the means to customize their scene composition, completely in design-time, without even running their games.

3.6 Playing the Simulation and Runtime

As the users' cutscenes play, Cine-AI listens to events fired when each time a shot marker is hit, executing the cinematography techniques during the marker's timestamp. However, even though almost all the data related to the camera motion is calculated in design-time, performing some of the operations in runtime is still necessary to allow better responses towards dynamic changes in the scene. This section explains the runtime aspect of Cine-AI.

3.6.1 Need for Runtime

The cinematography categories except *positioning* are responsible for camera behaviors that are performed continuously over multiple frames. For instance, the dolly zoom technique requires moving the camera away from the subject while simultaneously decreasing the camera's field of view. Calculating data that needs to be performed over multiple frames in design-time would only be eligible if users' cutscenes are completely static. However in practice, it is perfectly reasonable to have dynamic cutscenes, meaning that objects, animations and even the non-player character (NPC) behaviours during the cutscene might change depending on the game's state. For instance, an NPC character might walk towards two different directions during a cutscene, completely depending on the players' choices given during the previous gameplay. In this case, it would only make sense to perform the tracking technique implementation in runtime in order to account for various animation possibilities and any emergent changes regarding the 3D geometry. Thus, Cine-AI executes the *look*, *tracking* and *fx* categories during runtime.

3.6.2 Camera Collision Avoidance

Another important aspect in runtime camera motion is collision avoidance. Scene proxies partially handle the collision avoidance during design-time. However, solely using the proxies can not account for dynamic changes that would occur in 3D geometry during runtime. In order to overcome this problem, Cine-AI performs a runtime camera collision avoidance algorithm to make sure the camera does not clip any 3D geometry during the cutscene. Haigh-Hutchinson [15] presents a simple collision avoidance algorithm based on sliding the camera along the surface of the collided object. This procedure is applied in Cine-AI, as whenever the camera detects a collision during the cutscene, it starts sliding on the surface of the collided mesh with an angle perpendicular to the surface's normal. At some point, the camera will finish sweeping the surface, thus breaking the contact with the object. The sliding speed is dependent on the tracked object's velocity to perform immediate avoidance in high-speed cutscenes.

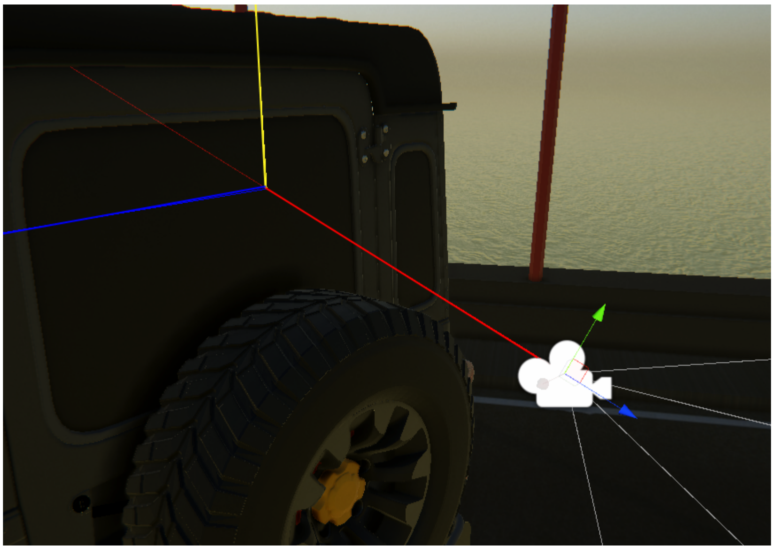

Figure 3.11: Camera performing line-casting for collision avoidance.

Chapter 4

Experiment 1: Video Comparison

To validate that Cine-AI is able to produce distinguishable director styles—at least in the case of our two directors—we conducted a single-session within-subjects study with 18 participants. Eight different cutscenes were prepared as the stimuli. In four of these cutscenes, Quentin Tarantino's sample data was used as an input, meanwhile the remaining four cutscenes were shot using Guy Ritchie's data. Participants were asked to watch the final results and assess which director's style each cutscene mimics. This study was designed to answer the following research questions:

- RQ1. How accurately do the users guess the right director?
- RQ2. Does the correct answer probability differ significantly from random chance?

4.1 Task Design

Since both of the sample directors focus on the action genre, all of the cutscenes were based on thrilling and fast-paced events with multiple characters involved. Some of the cutscenes were calmer than the others with steady dialogues between the characters, whilst some were completely based on action content such as shooting, chasing and fighting sequences [11]. The duration for the demo cutscenes ranged from thirty to sixty seconds.

Figure 4.1: A screen from one of the demo cutscenes, Port.

Additionally, we prepared two and a half minute long mashup videos for each of the sample directors, demonstrating their iconic cinematography and directing [11]. The mashup videos were prepared to be shown to the participants as references during the user study since no participant was required to have any previous affinity with the directors.

4.2 Participants and Recruitment Criteria

Exclusion criteria for the participants included having lack of sleep, being under the influence of drugs or alcohol, experiencing any digestive, muscle or organ pain or being under emotional distress. The eligibility criteria was that the participants have at least some affiliation with movies and video games.

18 participants were recruited by using call-outs and advertisements performed by the research group throughout online channels. In short answer format, gender and age were asked. Three of the participants reported being female (16%) and 15 of the participants reported being male (84%). Participant age ranged from 22 to 41 years old (M = 28.8, Mdn = 27.0, SD = 5.848).

4.3 Procedure

Following the recommendation of the Finnish National Board on Research Integrity, consent was obtained if the participant was at least 15 years old. Furthermore, additional consent was acquired due to the violent and gore nature of the reference clips and the cutscenes. After deeming eligible, participant was asked to fill out a demographics questionnaire. Subsequently, the participant was briefed about our toolset and it's purposes. The participant was informed about what a director style means in the context of our study. Information about camera management and director styles was given to better instruct how to approach watching the reference clips.

After the briefing, the participant was shown the reference clips of both directors and asked to analyze the shooting styles, camera management, main differences and possible signature techniques used in these clips. When the participant was ready to proceed, they were asked to watch the demo cutscenes in a random order and decide which director does each cutscene resemble the most. The reference clips for directors were available to re-watch any time in the form given to the participants. Finally a semi-structured interview was conducted in order to obtain information about the participant's confidence while deciding on the directors. This process was done by asking the participant to rate a number of statements about the procedure on 5-point Likert scale.

4.4 Hypothesis and Method

Since the participants are presented with a binary choice of two directors, our null-hypothesis for RQ2 was that classifying the videos is based on random chance and the probability of finding the correct director is 50%. A binomial test was used to test whether the null hypothesis should be rejected, pooling all the answers from all the participants, which results in a total sample size of 144. Such pooling is valid only if the repeated measures from each user can be assumed conditionally independent given the participant, which produces a data distribution similar to as if each kind of participant was recruited 8 times and each participant only provided a single guess. The assumption is valid here, as no feedback was provided about the correctness of the guesses and there should thus be no learning effects. Additionally, we analyzed inter-rater reliability to assess the subjectiveness of the director classification task.

4.5 Results

Demo	Description	Director	Correct %
Escape	Fight action	Tarantino	88.8
Meeting	Thriller	Tarantino	77.7
Facility	Spy action	Tarantino	77.7
Village	Drama	Tarantino	66.6
Bridge	Driving action	Ritchie	94.4
Alley	Foot chase	Ritchie	100
Night	Murder mystery	Ritchie	72.2
Port	Drama	Ritchie	55.5

Table 4.5: Cutscene information and the correct prediction percentages.

Individual results of the cutscenes used in our study are illustrated in Table 4.5. In total of 144 answers submitted by 18 participants, the total number of correct answers was 114 (79%, p = 0.791), which answers RQ1. The binomial test indicates that the null hypothesis of the data being random guesses should be rejected (p=9.9e-13). This provides evidence of a positive answer to RQ2: Cine-AI can produce recognizable director styles, at least in the case of our data and clips from two directors.

Further analysis of the results in Table 4.5 shows that some of the cutscenes were harder to predict than the rest, as the cutscenes *Village* and *Port* have the lowest success rate. This can be tied with the fact that both cutscenes were the slowest ones amongst others. They contained more dialogue and less action, making it difficult for an observer to spot the differences between the two directors. Since both of the sample directors rely on action and fast-paced content to demonstrate their iconic styles, these cutscenes contained less iconic cinematography styles but more common ones. One can assume that the success rate will drop as the genre of the cutscenes move further away from the sample directors' preferred styles. Moreover, the cutscene *Alley* can be considered as another outlier. All of the 18 participants were able to correctly predict that the *Alley* scene was shot using Guy Ritchie's data. We assume the reason for the agreement is that a particular chase sequence in the *Alley* scene was extremely similar to one of the scenes from the Guy Ritchie reference clip. All of the participants formed a direct connection between the scenes, some even reported that they have answered without even finishing the cutscene clip.

Semi-structured interview after the study revealed more information on the reliability of our procedure and the cutscenes. Most participants agreed that the difficulty of some scenes were correlated to the pace of the content within the scene. The general opinion was that the *Port* demo was the hardest one, as stated by P6, "due to the lack of shooting and fighting". Furthermore, majority of the participants agreed that the reference clips were crucially helpful on their decision process. However, three of the participants stated that these clips were guiding them too much, meaning that they make some of the cutscenes way too easy to decide. Participants generally enjoyed the cutscene content generated for the study and found them sufficient enough to tell a story. A great number of the participants stated that they can spot the effect of the Cine-AI and how it tries to mimic a particular director. On the other hand, three participants mentioned that the absence of more realistic characters, lip-sync and eye animations, as well as a well-mixed sound design made it harder to diagnose the scenes. The reason for the difficulty was that they tend to associate the director style with characters, emotion in the scene and music more than the camera work.

4.6 Inter-rater Reliability

As the reliability of an obtained result increases the research value [29], we performed Inter-rater Reliability (IRR) analysis on our results. As Kappa calculations are considered to be one of the most commonly used and original IRR methods [13], an extension of Kappa was chosen to calculate the reliability score of our study. Initially, Fleiss' Kappa, a multi-rater and chance-corrected [13] extension of Kappa calculations were used for IRR analysis. Fleiss' Kappa for 18 raters and 8 subjects yielded in the Kappa value of 0.382 ($z = 13.4$, p-value $= 0$). According to Landis and Koch [24], a Kappa value between 0.21 and 0.40 is considered fair, as shown in the table 4.6.

Value of K	Strength of Agreement %
≤ 0.01	Poor
0.01 - 0.20	Slight
0.21 - 0.40	Fair
0.41 - 0.60	Moderate
0.61 - 0.80	Substantial
0.81 - 1.0	Almost Perfect

Table 4.6:Kappa values interpretation according to Landis and Koch [24].

The results indicate that our test procedure has available room for improvement, as a lower inter-rater reliability means lower chance of acquiring reliable data. However, it has also been mentioned in the literature that

Fleiss' Kappa can lead to paradoxical results [14], scoring differently when the order of data-set changes or penalizing the score when there is a high rate of observed agreement. Warrens [37] provides a formal proof of the same problem in Cohen's Kappa, another extension of Kappa calculations. Despite the existence of numerous research to remove these paradoxes [8, 12], it is not uncommon for studies include more than one measures of reliability testing.

Another significant inter-rater reliability test is Intra-class Classification (ICC) analysis. ICC quantifies IRR based on the magnitude estimates of rater disagreements [16] and the correlation between different raters [23]. Since each cutscene was assessed by each participant in our study and the participants were the only raters of interest, we have chosen a fixed raters version of the ICC analysis. Upon calculating the reliability from a mean of k raters measurement (k = 18, ICC3k [34]), the analysis yielded in a value of 0.93 (*p-value* = 5.0e-13) with a confidence interval of 0.85 (lower bound) and 0.98 (upper bound). Koo and Li [23] state that using the lower and upper bound values is recommended while estimating the categorization of the results. Thus, amongst the reliability classification of poor ($icc \leq 0.5$), moderate ($0.45 < icc \leq 0.75$), good ($0.75 < icc \leq 0.9$) and excellent ($icc > 0.9$), our results yield between *good* and *excellent* reliability.

Chapter 5

Experiment 2: Tool Interaction

While Experiment 1 provides a sanity check for our technical approach, it provides no information about the value of Cine-AI for game developers. Our purpose was to assess usability as well as to get feedback on various aspects of our toolset. More specifically, the study aimed to answer the following research questions:

- What is the overall usability of the toolset?
- Which aspects of the toolset work as intended, and which aspects do not?
- Are there any major flaws regarding usability and user experience?
- Are there any fatal errors that prevent the users from running the toolset successfully?
- How should one improve the toolset?

In order to assess the usability, we have used System Usability Scale (SUS), a standardized usability test proven to be quick and reliable way of measuring a usability on an easily interpretable scale from 0 to 100% [25].

5.1 Task Design

Since our toolset was implemented in Unity 3D game engine, we have developed a simple cutscene in order to act as a testbed for the user study. The scene included various characters having a dialogue and walking around in an environment representing a living room. The demo scene did not have any gameplay, but only the cutscene animations ready to be played, with no camera motion attached to it. We presented this scene to the users and asked them to start using Cine-AI, adding shot markers and simulating them in order to produce a version of the cutscene usable in a game.

5.2 Inclusion Criteria and Participants

The primary eligibility criteria for participating was that the participants must have prior experience using Unity 3D game engine, since Cine-AI was implemented in it. The participants had to be comfortable navigating the Unity 3D user interface, being able to effectively use the Timeline tool the game engine provides and be able to test their work both in editor and also in game mode without any problems. 12 participants were recruited, 3 of the participants were friends of an author, four were students at Aalto University and others were recruited from online Unity 3D channels. A short demographics questionnaire revealed that three participants reported being female (25%) and nine participants reported being male (75%). Participants included professional animators and 3D artists, as well as Unity developers.

5.3 Procedure

The study was conducted in three distinct stages. First stage was the briefing stage in which the users were introduced to the toolset, the purpose of the research and the capabilities of Cine-AI. They were shown windows and the required asset files in order to setup and use the toolset. Secondly, users were asked to start using the toolset by going through the initial setup stage. They were asked to import the sample director data, define scene proxies and run the simulation, as well as to tweak the results to best fit their needs. This stage was observed by an author and all participants were asked to perform Concurrent Think-Aloud Protocol (CTA). Since CTA has been proven to be 80% accurate way of capturing the participants' actual thought processes [9], we have expected it to help us analyze the major usability issues in our toolset. During the process, the author took notes of any minor and major thoughts and comments related to the usability of the toolset. Upon completion of the demo scene simulation, users were asked to fill in a SUS questionnaire. This was followed by a semi-structured interview between the participant and the author in order to find out user preferences and receive additional feedback.

5.4 Results

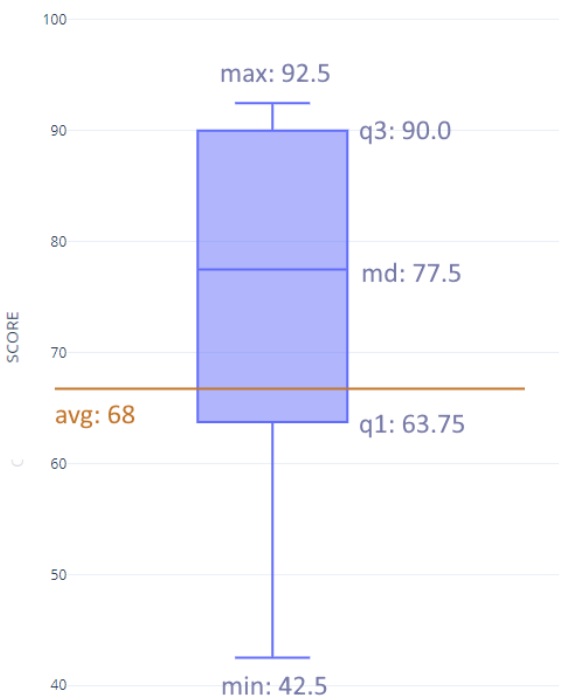

Figure 5.1: Orange line demonstrates the population average for SUS scores [32]. Cine-AI has scored above average with maximum of 92.5 out of 100.

Based on the SUS calculation rules [3], a SUS score for each question was calculated. Cine-AI has scored 74.375 out of 100 by the study conducted with 12 participants ($MD = 77.5$, $SD = 17.50$). Cine-AI's score results in a percentile rank of 70% [32]. Thus, the result indicates that Cine-AI performs above average and better than seven out of ten other systems. According to a curved grading scale interpretation from A to F [33], our score corresponds to a grade of B.

No critical or severe usability issues were reported throughout CTA procedures. All of the participants were able to continue working through the cutscene task they were given without any major problem. However, participants mostly agreed on the lack of tooltips and stated that they require more information in order to understand what each adjustable parameter means. Generally participants liked the storyboard window, stating that it looks professional and helps them visualize their composition better. Furthermore, some participants suggested better layout options for the storyboard window and the ability to use multiple aspect ratios for shot previews. Three participants complained about the setup procedure of the toolset and one requested a wizard to automate setup steps. There were also additional requests based on the game engine we have implemented the toolset in, Unity 3D. Two of the participants stated that they would prefer if the toolset was completely integrated within Unity's own tools. They suggested features like animation baking into Unity's timeline or runtime editing support, which we have tried to avoid in the scope of this research to prevent strong coupling with any game engine. In overall, participants reported minor improvements such as tooltips, helper windows, better looking tools and in-tool documentation, but they were able to enjoy the toolset.

Chapter 6

Discussion

Our experiments indicate that Cine-AI works as expected and can help developers and artists to compose their scenes using a target director's shooting style. Our experiment comparing the generated results to actual director clips indicates that our toolset is able to accurately mimic the sample director's cinematography styles with a fair amount of confidence. The usability study we have conducted shows that user interface of Cine-AI is sufficient enough to allow the usage of the toolset without any major flaws.

Imitating a director style is a process that includes a lot of subjective elements as well as considerable amount of variance in the input data, as no two director can be considered the same. Naturally, it would not be adequate to expect perfect precision on such a task. Thus, we do not expect Cine-AI to replace the process of manual directing. On the other hand, Cine-AI can be used as a baseline to achieve the director's style. Users, especially independent developers, can use Cine-AI to invoke ideas about how their favourite director would shoot their cutscene. Additionally, Cine-AI would also handle most of the ground work by automating the camera management process. Developers can base their initial scene on the results of Cine-AI, which can be extended upon by manual editing.

It is also possible to extent Cine-AI into a product that can be used in triple-A game industry. The main obstacle for such an extension would be the analysis of the sample director data. It is possible to use sophisticated machine learning techniques in order to automate the process of watching director clips. If one is able to train a machine learning system to differentiate various cinematography techniques from a sequence of images, it would be possible to rapidly produce distinct director data and categorize them into different use-cases for possible game scenarios. Having multitude of director data samples, Cine-AI would be able to produce at least initial results for scene directing and might remove the need of hiring an actual director.

Chapter 7

Limitations and Future Work

The most crucial part of our research is the processing and implementation of the director data. 80 different movie clips per director in the case of our study were analyzed by the authors. The analysis of the director data was inevitably biased due to the fact that it was limited with the cinematography knowledge of the authors. In future work, it should be possible to reduce the bias by using multiple coders work on the same director. Thus, it would be possible to conduct statistical analysis on the director data in order to increase the reliability. Furthermore, our results heavily rely on the sample directors we have chosen. Quentin Tarantino and Guy Ritchie are well known for their distinctive and iconic styles, and are good sources of sampling in regards to recognizing various cinematography styles. However, not all directors solely focus on the camera work to express their practice. For instance, Tim Burton, a well-accomplished director known for his gothic style, achieves his unique results mostly through dark and edgy character design [5]. Such cases might occur, where the directors tend to use common and non-distinguishable camera techniques but accomplish their signatures through other means. In such cases, it would not be reasonable to expect Cine-AI to work accurately. In future work, it is possible to incorporate more aspects of film making into Cine-AI, such as more information about the characters and entities in the cutscene, post editing features and visual effects. By encoding director data for other aspects of film making and implementing these aspects in Cine-AI, it is possible to significantly increase the accuracy of the imitation process.

In a movie production pipeline, the work and responsibilities of a director and cinematographer are clearly separated [4]. Cinematography techniques, including any decisions regarding the digital setup and the camera work mostly belong to the cinematographer. However, cinematographers work under the directors' influence and it is common to say that the resulting work, including any choice regarding the cinematography techniques, are heavily influenced by the directors' decisions and styles [36]. Thus, we did not make the distinction of a director and a cinematographer in our study. It is possible to further analyze the distinction and incorporate the differences into Cine-AI. One suggestion would be to create multiple aspects of the toolset for separate purposes such as cinematography setup and directing, which would increase the audience of the toolset by providing support for more aspects of film making.

In the current state of our toolset, simulation of the timeline markers results in a linear scene composition, where each marker will have a specific cinematography technique assigned to it for each of the technique categories. Therefore, there would only exists a single transition from marker A to B and the users have to re-simulate the markers if they desire to change the results. In future work, we plan to explore the storyboard implementation more comprehensively, possibly allowing multiple options for transitions. Technically, it is possible to simulate a number of possible outcomes for the marker i by looking at the results of the marker $i-1$. Therefore, it would be possible to provide various transition and shot suggestions suggestions to the users. Lino et al. [27] introduce a ranking system for possible shot compositions, based on the notion of a *screenplay*. Screenplay is a structure containing information about a shot, including the scene information, involved actors and their 3D object data, as well as the textual descriptions of the actions performed by the actors. The screenplay data is used to compute suggestions for automated camera motion. We plan to implement a similar system, where our system will generate multiple shots based on different cinematography techniques and rank them by using the previous shot marker's screenplay data. Therefore, it will be possible to create a more professional toolset ready for advanced production pipelines.

We plan to extend the functionality of the storyboard window in Cine-AI by increasing the number of automated features in our methodology. For instance, Cine-AI includes hard-coded rules of cinematography, such as the rule of thirds or Arijon [2]'s idioms. Instead of relying on particular sets of rules, Cine-AI can be improved to automate the process of rule filtering. For instance, a cinematography rule set in the format of a formal language such as DCCL [6] can be implemented. Thus, users would be able to define their own rules and Cine-AI would automatically try to abide by them while selecting possible shot techniques and configurations.

Following the suggestions made by our participants in the usability study, we plan to improve the user interface of Cine-AI to provide more intuitive and accessible features. Even though we would like to keep Cine-AI decoupled from any game engine as much as possible, we had received great feedback to implement more features in Unity 3D game engine that would increase the toolset's performance and usability. Therefore, in future work we plan to implement the said feature requests, possibly as a forked version of Cine-AI, to see and measure how the tools provided by Unity would improve our system's quality.

Chapter 8

Conclusion

We have presented Cine-AI, a novel tool that incorporates the characteristics of movie directors' styles with a procedural camera generation algorithm for in-game cutscenes. Our methodology allows the users to procedurally direct their cutscenes while imitating the cinematography style of a target director. Cine-AI's interface provides extensive tools for adjusting simulation properties and tweaking the final scene composition. Our video comparison study shows the participants were able to accurately predict which director was being emulated by the cutscenes generated with Cine-AI. SUS data indicates that Cine-AI provides an above average usability. Despite having room for improvement and minor usability issues, Cine-AI provided an environment where participants were able to use the toolset and complete the demo tasks without any major technical flaws.

The main novelty of Cine-AI is the process of using an actual director's cinematography data—extracted through annotating a dataset of example videos—as an input to generate shot configurations and scene compositions. Cine-AI also combines a highly configurable design-time workflow with run-time support for emergent changes. Our open-source implementation of Cine-AI in Unity 3D game engine allows a large audience to experiment with our toolset. To our knowledge, no previous system provides autonomous cinematography composition and camera management and also incorporates empirical data of a director for imitating a particular style.

Bibliography

[1] Dan Ablan. *Digital cinematography & directing*. New Riders, 2002.

[2] Daniel Arijon. Grammar of the film language. In *Grammar of the Film Language*. Silman-James Press ; Distributed by Samuel French Trade, 1991.

[3] John Brooke. Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, 189, 11 1995.

[4] Blain Brown. *The Basics of Filmmaking: Screenwriting, Producing, Directing, Cinematography, Audio, & Editing*. Routledge, 2020.

[5] Tim Burton. *Burton on Burton*. Faber & Faber, 2008.

[6] David Christianson, Sean Anderson, Li-wei He, David Salesin, Daniel Weld, and Michael Cohen. Declarative camera control for automatic cinematography. *Proceedings of the National Conference on Artificial Intelligence*, 1, 12 1996.

[7] Marc Christie and Patrick Olivier. Camera control in computer graphics: Models, techniques and applications. 01 2009. doi: 10.1145/1665817.1665820.

[8] Domenic V. Cicchetti and Alvan R. Feinstein. High agreement but low kappa: Ii. resolving the paradoxes. *Journal of Clinical Epidemiology*, 43(6):551–558, 1990. ISSN 0895-4356. doi: https://doi.org/10.1016/0895-4356(90)90159-M. URL `https://www.sciencedirect.com/science/article/pii/089543569090159M`.

[9] L. Cooke. Assessing concurrent think-aloud protocol as a usability test method: A technical communication approach. *IEEE Transactions on Professional Communication*, 53(3):202–215, 2010. doi: 10.1109/TPC.2010.2052859.

[10] S. Drucker and D. Zeltzer. Intelligent camera control in a virtual environment. 1994.

[11] Inan Evin. Director mashups and demo cutscenes for user experiment: Video comparison, 2021. URL `https://www.youtube.com/watch?v=sPRTwby-1fU&list=PLOpXSM_412yrRipj3OdWy5hSvoWarA-IT`.

[12] Rosa Falotico and Piero Quatto. Fleiss' kappa statistic without paradoxes. *Quality Quantity*, 49:463–470, 03 2014. doi: 10.1007/s11135-014-0003-1.

[13] Natasa Gisev, J. Simon Bell, and Timothy F. Chen. Interrater agreement and interrater reliability: Key concepts, approaches, and applications. *Research in Social and Administrative Pharmacy*, 9(3):330–338, 2013. ISSN 1551-7411. doi: https://doi.org/10.1016/j.sapharm.2012.04.004. URL `https://www.sciencedirect.com/science/article/pii/S1551741112000642`.

[14] Kilem Gwet. *Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters*. 01 2012.

[15] Mark Haigh-Hutchinson. Chapter 9 - motion and collision. In Mark Haigh-Hutchinson, editor, *Real Time Cameras*, pages 313–364. Morgan Kaufmann, Boston, 2009. ISBN 978-0-12-311634-5. doi: https://doi.org/10.1016/B978-0-12-311634-5.00009-4. URL `https://www.sciencedirect.com/science/article/pii/B9780123116345000094`.

[16] Kevin A Hallgren. Computing inter-rater reliability for observational data: An overview and tutorial. *Tutorials in quantitative methods for psychology*, 8 1:23–34, 2012.

[17] Shakil A. Hannan, Daisuke Endou, and Masanobu Yamamoto. Director-oriented shot assessment and evaluation in virtual cinematography. *The Journal of the Institute of Image Information and Television Engineers*, 59(4):592–603, 2005. doi: 10.3169/itej.59.592.

[18] Li-wei He, Michael Cohen, and David Salesin. The virtual cinematographer: A paradigm for automatic real-time camera control and directing. pages 217–224, 01 1996. doi: 10.1145/237170.237259.

[19] Timothy Heiderich. Cinematography techniques: The different types of shots in film. *Videomakers. Accessed September*, 22:2020, 2012.

[20] Arnav Jhala and R. Young. A discourse planning approach to cinematic camera control for narratives in virtual environments. volume 1, pages 307–312, 01 2005.

[21] Peter Karp and Steven Feiner. Automated presentation planning of animation using task decomposition with heuristic reasoning. In *Proceedings of Graphics Interface '93*, GI '93, pages 118–127, Toronto, Ontario, Canada, 1993. Canadian Human-Computer Communications Society. ISBN 0-9695338-2-9. URL `http://graphicsinterface.org/wp-content/uploads/gi1993-15.pdf`.

[22] Kevin Kennedy and Robert Mercer. Planning animation cinematography and shot structure to communicate theme and mood. *ACM International Conference Proceeding Series*, 22:1–8, 01 2002. doi: 10.1145/569005.569006.

[23] Terry K. Koo and Mae Y. Li. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *Journal of Chiropractic Medicine*, 15(2):155–163, 2016. ISSN 1556-3707. doi: https://doi.org/10.1016/j.jcm.2016.02.012. URL `https://www.sciencedirect.com/science/article/pii/S1556370716000158`.

[24] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[25] James R. Lewis. The system usability scale: Past, present, and future. *International Journal of Human–Computer Interaction*, 34(7):577–590, 2018. doi: 10.1080/10447318.2018.1455307. URL `https://doi.org/10.1080/10447318.2018.1455307`.

[26] Christophe Lino, Marc Christie, Fabrice Lamarche, Schofield Guy, and Patrick Olivier. A Real-time Cinematography System for Interactive 3D Environments. In *SCA '10 Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 139–148, Madrid, Spain, July 2010. doi: 10.2312/SCA/SCA10/139-148. URL `https://hal.archives-ouvertes.fr/hal-00535865`.

[27] Christophe Lino, Marc Christie, Roberto Ranon, and William Bares. The director's lens: an intelligent assistant for virtual cinematography. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 323–332, 2011.

[28] Joseph Mascelli. *The Five C's of Cinematography: Motion Picture Filming Techniques*, pages –251. 10 1998. ISBN 187950541X.

[29] M. McHugh. Interrater reliability: the kappa statistic. *Biochemia Medica*, 22:276 – 282, 2012.

[30] Thomas Oskam, Robert Sumner, Nils Thürey, and Markus Gross. Visibility transition planning for dynamic camera control. page 325, 11 2010. ISBN 978-3-642-16957-1. doi: 10.1007/978-3-642-16958-8_30.

[31] Rémi Ronfard, Vineet Gandhi, and Laurent Boiron. The prose storyboard language: A tool for annotating and directing movies. *CoRR*, abs/1508.07593, 2015. URL `http://arxiv.org/abs/1508.07593`.

[32] Jeff Sauro. Measuring usability with the system usability scale (sus), 2011. URL `https://measuringu.com/sus/`.

[33] Jeff Sauro and James R. Lewis. Chapter 8 - standardized usability questionnaires. In Jeff Sauro and James R. Lewis, editors, *Quantifying the User Experience*, pages 185–240. Morgan Kaufmann, Boston, 2012. ISBN 978-0-12-384968-7. doi: https://doi.org/10.1016/B978-0-12-384968-7.00008-4. URL `https://www.sciencedirect.com/science/article/pii/B9780123849687000084`.

[34] P. Shrout and J. Fleiss. Intraclass correlations: uses in assessing rater reliability. *Psychological bulletin*, 86 2:420–8, 1979.

[35] Edirlei Soares de Lima, Cesar Pozzer, Marcos d'Ornellas, Angelo Ciarlini, Bruno Feijó, and Antonio Furtado. Virtual cinematography director for interactive storytelling. pages 263–270, 10 2009. doi: 10.1145/1690388.1690432.

[36] David Stump. *Digital cinematography: fundamentals, tools, techniques, and workflows*. CRC Press, 2014.

[37] Matthijs Warrens. A formal proof of a paradox associated with cohen's kappa. *Journal of Classification*, 27(3):322–332, 2010. URL `https://EconPapers.repec.org/RePEc:spr:jclass:v:27:y:2010:i:3:p:322-332`.

[38] Paul Wheeler. *Practical cinematography*. Taylor & Francis, 2005.

[39] R. Young and Johanna Moore. Dpocl: A principled approach to discourse planning. 03 2001.